

DHB:vjs 04/13/07 682672 303993.01
PATENT

Attorney Reference Number 3382-64711-01
Application Number 10/602,538

Amendments to the Claims

1. A method comprising:
executing an exchange function comprising exchanging an address contained in a local variable with an address in a global variable for an executing first requestor, wherein there are plural requestors in contention for the shared resource;
jumping to the address in the local variable wherein the address in the local variable points to the exchange function address when one of the plural requestors is accessing the shared resource or to a shared resource address when the shared resource is not otherwise being accessed; such that a single requestor uses the shared resource at a time; and
executing the exchange function after accessing the shared resource.
2. The method of claim 1 wherein the exchange function is a method of a class including a property comprising the global variable.
3. The method of claim 1 wherein the plural requestors are plural multi-tasking processes.
4. The method of claim 1 wherein the plural requestors are plural time-sharing threads.
5. (Canceled)
6. The method of claim 1 wherein the local variable is associated with a process.
7. The method of claim 1 wherein the local variable is associated with a thread.
8. The method of claim 1 wherein the local variable is a property of a class associated with a thread.

DHB:vjs 04/13/07 682672 303993.01
PATENT

Attorney Reference Number 3382-64711-01
Application Number 10/602,538

9. The method of claim 1 wherein the exchange function is a method of a class including a property comprising the local variable.
10. (Canceled)
11. The method of claim 1 wherein the plural requestors are plural child threads of a process in contention for the shared resource.
12. The method of claim 11 wherein the child threads each include a local variable, and the process includes the global variable.
13. The method of claim 1 wherein the exchange function is executed as an atomic unit.
14. A computer-readable medium comprising instructions executed by a processor to perform the method of claim 1.
15. (Currently amended) A computer-readable medium including instructions executed by a processor to:
- create a global variable initialized with an address of a shared resource;
 - create plural threads each including a local variable initialized with an address of an exchange function;
 - the exchange function being called by the plural threads to exchange an address in a calling thread's local variable with an address in the global variable;
 - after the calling thread calls the exchange function, transferring control to an address in the calling thread's local variable wherein the address in the local variable points to the address of the exchange function or to the address of the shared resource; and
 - the exchange function being called by the calling thread after the calling thread accesses the shared resource;

DHB:vjs 04/13/07 682672 303993.01
PATENT

Attorney Reference Number 3382-64711-01
Application Number 10/602,538

wherein the plural threads are in contention for the shared resource, and wherein the address in the calling thread's local variable points to the shared resource when the shared resource is not otherwise being accessed, such that one thread uses the shared resource at a time.

16. The computer-readable medium of claim 15 wherein the function for creating the global variable further comprises creating as an atomic unit.

17. The computer-readable medium of claim 15 wherein the shared resource is a memory location.

18. The computer-readable medium of claim 15 wherein the exchange function further comprises calling and returning from a sleep function before exchanging variables.

19. The computer-readable medium of claim 15 wherein transferring control to the address in the thread's local variable transfers control to one of the exchange function or the shared resource; and
execution of the thread continues at the transferred location.

20. (Currently amended) A computer system comprising:
a processor coupled to memory comprising a global variable;
a shared resource;
the global variable initialized with an address of the shared resource;
an exchange function for exchanging contents of the global variable with contents of local variables;
plural units of execution each comprising:
an associated local variable initialized with an address of the exchange function,
code for calling the exchange function to exchange contents of the global variable with contents of a local variable,
code for, after calling the exchange function, jumping to an address contained in its associated local variable wherein the address points to the address of the exchange function or to the address of the shared resource, wherein the address points to the

DHB:vjs 04/13/07 682672 303993.01
PATENT

Attorney Reference Number 3382-64711-01
Application Number 10/602,538

address of the shared resource only when the shared resource is not otherwise being accessed, and
code for executing the exchange function after accessing the shared resource.

21. The computer system of claim 20 wherein said plural units of execution are threads, and a first thread includes code for, after executing the exchange function, jumping to the address accessing the shared resource, and before the first thread exits the shared resource, the computer system includes code for transferring control of the processor to a second thread, and the second thread includes code for, upon executing the exchange function, jumping to the address of and cycles in the exchange function.

22. The computer system of claim 21 wherein after control transfers to the first thread, and after the first thread exits the shared resource and executes the exchange function, and after control of the processor transfers to the second thread, the second thread includes code for executing the exchange function and jumping to the address accessing the shared resource.

23. (Previously Presented) The computer system of claim 20 wherein the exchange function is a method of a class including a property comprising the global variable.

24. (Canceled)

25. (Canceled)

26. (New) The method of claim 18 wherein the sleep function waits a variable time.

27. (New) The method of claim 20 wherein the code for executing the exchange function after accessing the shared resource allows another unit of execution to access the shared resource.

28. (New) The method of claim 20 wherein the local variable address points to the address of the exchange function when the shared resource is being accessed.